

Fast Mode Decision for H.264/AVC Based on Local Spatio-Temporal Coherency

Wei Zhang, Yan Huang and Jingliang Peng*

School of Computer Science and Technology, Shandong University
Shandong Provincial Key Laboratory of Software Engineering, China

* Corresponding author(Email: jpeng@sdu.edu.cn)

Abstract—In this paper, we propose a fast mode decision algorithm for H.264/AVC. It is based on the spatio-temporal coherency of the local neighborhood including and around the current block. We first build the histograms of the current block and the co-located block in the reference frame, respectively. If the difference between the two histograms is small, we use large-block-size modes. Otherwise, we subdivide the current block into four equal-sized sub-blocks and estimate the motion vector (MV) for each. In general, if there is a high degree of coherency between those MVs, we use large-block-size modes and otherwise small-block-size modes. In addition, we use the number of neighboring large blocks and the sub-blocks' rate-distortion (R-D) costs as further hints for the mode decision. As experimentally demonstrated, our algorithm leads to significant saving in computing time on the test video sequences.

I. INTRODUCTION

With the development of digital video technology, the demands of people to get higher-quality video services and more enjoyable visual experience are rapidly increasing. To meet these demands, the JVT H.264 video coding standard [1] was proposed, which undoubtedly has made a prominent progress beyond the earlier video coding standards. It provides higher compression ratios without visual quality decrease and better network compatibility as well. However, it demands much more on computational power. In general, mode decision and motion estimation will take about 90% of the overall encoding time [2]. Hence many researchers have focused on how to reduce the computational complexity of mode decision with many fast mode decision algorithms proposed [3], [4], [5], [6], [7], [8].

In [3], Wang *et al.* proposed to jointly optimize mode decision and motion estimation. With the help of theoretical analysis, the method checks a set of conditions to early terminate the mode decision and the motion estimation. In [4], Jing and Chau proposed a fast mode decision method dependent only on the absolute differences between consecutive frames. In their method, the current frame was classified as homogeneous regions or not by the mean absolute frame difference of the current frame and the mean absolute difference of the current block. In [5], Wu *et al.* proposed a fast inter-mode decision algorithm which makes use of the spatial homogeneity and the temporal stationarity characteristics of video objects. The spatial homogeneity of a block is decided by that block's edge intensity, and the temporal stationarity is decided by

the difference of that block and its co-located counterpart in the reference frame. In [6], Kim proposed an algorithm based on temporal correlation for P-slices. This algorithm uses a simple block tracking scheme with a P-16x16 block type in the previous frame to get the R-D cost of the most correlated block and the R-D cost was used to early terminate the mode search. In [7], Pi *et al.* proposed an inter-mode decision scheme. It predicts the current block's best mode from the neighboring blocks using spatio-temporal correlation and estimates its R-D cost from its co-located block in the previous frame. In [8], Zeng *et al.* proposed a fast mode decision algorithm based on motion activity. This algorithm starts with checking the R-D cost computed at the SKIP mode for a possible early termination. Depending on the R-D cost, different modes will be chosen. For the condition that the R-D cost is between a 'high' threshold and a 'low' threshold, the remaining seven modes would be classified into three motion activity classes and separately examined.

In this paper, we propose a novel fast mode decision algorithm based on the local spatio-temporal coherency of the local neighborhood including and around the current block. In general, the higher (lower) the temporal and/or the spatial coherency is in the local neighborhood, the more probably the large-block-size modes (small-block-size modes) are the best choice. Primarily based on this observation, we design our algorithm that leads to significantly reduced computation while yielding good rate and distortion performance at the same time.

The rest of this paper is organized as follows. In Section II, we describe the proposed algorithm in detail; in Section III, we present and discuss the experimental results; in Section IV, we conclude this work.

II. THE PROPOSED MODE DECISION ALGORITHM

A. Observations

In the H.264/AVC reference software JM 17.2 [9], all candidate modes will be checked using the Lagrangian rate-distortion optimization (RDO) function. Then the mode which results in the least R-D cost will be chosen as the best mode. Although the exhaustive mode decision algorithm is the most precise one, its computational complexity is extremely high. Acceleration will be achieved if we skip some modes during

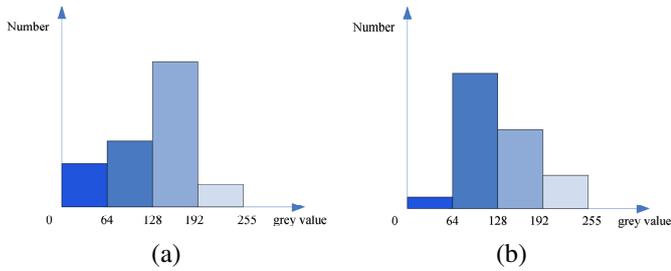


Fig. 1. Histograms of (a) the current block and (b) the co-located block in the reference frame.

the mode decision process, and our acceleration algorithm is specifically based on the following observations.

- The difference between the current block's and its co-located block's illumination characteristics is closely related to the current block's mode. Typically, if the gray level histogram of the current block is similar to that of its co-located block in the previous frame, the current block is more likely to have a large-block-size mode.
- We find that there is a close relation between the motions of the current block's sub-blocks and the current block's mode. If the four 8×8 sub-blocks' motion vectors are the same or almost the same, the current block tends to have a large-block-size mode.
- Utilizing the inter-block spatial correlation to further prune unnecessary modes is often an effective method. For instance, the modes of the spatially neighboring blocks provide further hints on the current block's candidate modes.

B. The Proposed Algorithm

We first build the histograms of the current block and its co-located block in the previous frame, examples of which are shown in Fig. 1 (a) and Fig. 1(b), respectively. We quantize the whole range of gray levels to four bins: $[0, 63]$, $[64, 127]$, $[128, 191]$ and $[192-255]$. Denoting the two histograms as A and B , $A(i)$ ($i=0,1,2,3$) and $B(i)$ gives the number of pixels with gray levels in the i -th bin for the two histograms, respectively. We compute the difference, D_{hist} , between histograms A and B and conduct a double-thresholding on D_{hist} . Specifically, if $D_{hist} \leq D_{low}$, we make the decision between SKIP or $INTER_{16 \times 16}$ according to their R-D costs and skip all the other modes; if $D_{low} < D_{hist} \leq D_{high}$, we make the decision between SKIP, $INTER_{16 \times 16}$, $INTER_{16 \times 8}$, $INTER_{8 \times 16}$ according to their R-D costs and skip all the other modes; otherwise, we continue with the following process.

We subdivide the 16×16 block into four 8×8 sub-blocks and perform the motion estimation operation for each individually. As a result, we obtain four motion vectors $\{MV_0, MV_1, MV_2, MV_3\}$, as shown in Fig. 2.

Observing that high coherency between the sub-blocks' motions often implies large-block-size modes, we reduce the candidate modes by thresholding on the similarity between the sub-blocks' MVs. Specifically, the coherency between the sub-blocks is inversely related to the difference between the MVs,

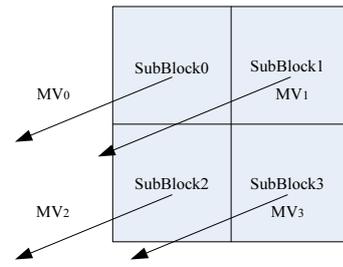


Fig. 2. Equal-sized sub-blocks of the current block.

$Diff_4(MV_0, MV_1, MV_2, MV_3)$ or $Diff_2(MV_i, MV_j)$, $i, j \in \{0, 1, 2, 3\}, i \neq j$. Thereafter, if the current block has high sub-block motion coherency, we will consider large-block-size modes with a higher priority; otherwise we consider small-block-size modes and may also consider large-block-size modes only if the neighboring large block count(NLBC) is above a threshold.

The detailed algorithm is given in Algorithm 1.

C. Block histogram

In order to compare the illumination difference between two blocks, we compute and compare their gray level histograms. For efficiency considerations, we build and compare coarse histograms of blocks. Specifically, we divide the gray level range of $[0, 255]$ to four equally sized bins: $[0, 63]$, $[64, 127]$, $[128, 191]$ and $[192-255]$ and count the number of pixels in each bin for each block. The test videos in our experiments are all stored in the YUV format, and we obtain the gray level information directly from the Y channel. The difference, D_{hist} , of two histograms, A and B , is computed as

$$D_{hist} = \sum_{i=0}^3 |A(i) - B(i)|. \quad (1)$$

Fig. 3 illustrates the close relation between a block's mode and its difference from its counterpart in the previous frame. Fig. 3(a), Fig. 3(b) and Fig. 3(c) show the 11th frame from football(cif), the pixelwise difference between the 11th and the 10th frames and the best mode of each block in the 11th frame, respectively. From Fig. 3(b) and Fig. 3(c), we observe that a block with a small inter-frame difference tends to have a large-block-size mode. The example shown in Fig. 3 is typical of many commonly used videos. As such, this observation provides a solid support to our approach of histogram-difference-based large-block-size mode prediction.

To further corroborate the effectiveness of our histogram-difference-based large-block-size mode prediction, we give in Table I the prediction accuracy for blocks with $D_{hist} \leq D_{low}$ and $D_{low} < D_{hist} \leq D_{high}$, respectively, for several test video sequences. We get the value of D_{low} and D_{high} by repeated experiments. From this table, we observe hit ratios of well above 90% for all the test sequence.

D. The Difference of MVs

Through our observation, we find that we may predict the best mode through the degree of coherency between the four

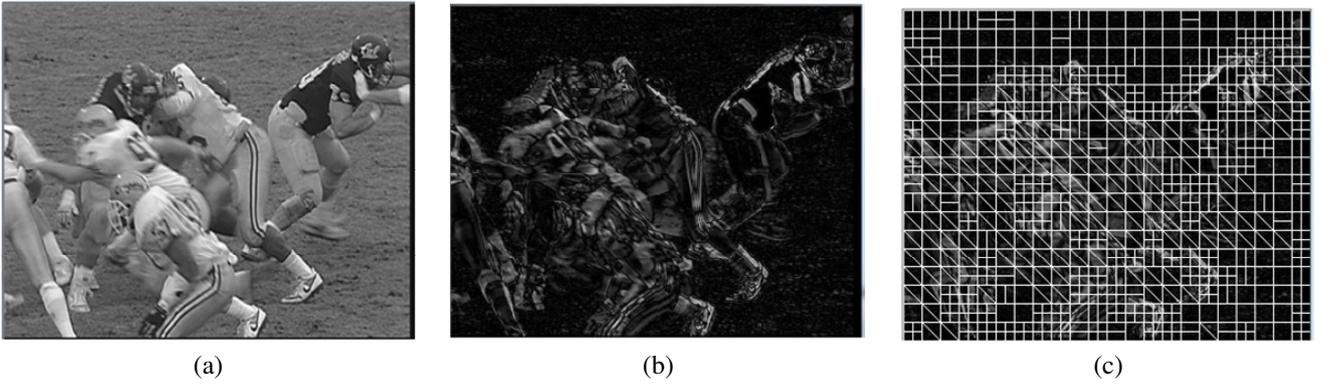


Fig. 3. (a) The 11th frame from football(cif), (b) the pixelwise difference between the 10th and the 11th frames, (c) the mode types of the blocks in the 11th frame.

TABLE I
THE ACCURACY OF LARGE-BLOCK-SIZE MODE PREDICTION BASED ON THE D_{hist} .

sequence	$D_{hist} \leq D_{low}$ (%)	$D_{low} < D_{hist} \leq D_{high}$ (%)
foreman(cif)	97.1	97.3
mobile(cif)	99.0	99.6
flower(cif)	99.4	99.8
akiyo(qcif)	99.4	99.3
hall(qcif)	99.7	94.4

motion vectors, MV_0, MV_1, MV_2, MV_3 . Specifically, we differentiate various configurations based on the differences between the MVs. We use $Diff_4(MV_0, MV_1, MV_2, MV_3)$ and $Diff_2(MV_i, MV_j)(i, j \in \{0, 1, 2, 3\}, i \neq j)$ to measure the difference between four and two MVs, respectively. They are computed as follow:

$$Diff_4(MV_0, MV_1, MV_2, MV_3) = |MV_0 - MV_1| + |MV_2 - MV_3| + |MV_0 - MV_2| + |MV_1 - MV_3|. \quad (2)$$

$$Diff_2(MV_i, MV_j) = |MV_i - MV_j|, \quad (i, j \in \{0, 1, 2, 3\}, i \neq j). \quad (3)$$

We find that the four sub-blocks' motion vectors often have small differences when the block's mode is SKIP or $INTER_{16 \times 16}$. Otherwise, two of the motion vectors tend to have small differences when the block's mode is $INTER_{16 \times 8}$ or $INTER_{8 \times 16}$. The above observation forms the basis of the latter part of Algorithm 1, in which, we test the following conditions:

$$Diff_4(MV_0, MV_1, MV_2, MV_3) < T_4 \quad (4)$$

$$Diff_2(MV_0, MV_1) < T_2 \text{ or } Diff_2(MV_2, MV_3) < T_2 \quad (5)$$

$$Diff_2(MV_0, MV_2) < T_2 \text{ or } Diff_2(MV_1, MV_3) < T_2 \quad (6)$$

In these conditions, the values of T_2 and T_4 should be discriminative for the block mode prediction. In our paper, we empirically determine their values.

When any of the above conditions is met, we predict the best mode to be a large-block-size mode and otherwise a small-block-size mode. Denoting the above three conditions as C_1, C_2 and C_3 , the corresponding statistical data for mode prediction accuracy are given in Table II. From this table, we see that the prediction for large-block-size modes is in general more accurate than for small-block-size modes. But the prediction accuracy for large-block-size modes is still not high enough corresponding to C_2 or C_3 . In order to improve the prediction accuracy for large-block-size modes, we further check if the R-D costs of the large-block-size modes are bigger than those of the small-block-size modes. If they are, small-block-size modes may be the accurate choices instead, and we also include the small-block-size modes as candidates. This is achieved by the comparison of $RDCost_{8 \times 8}$ and $RDCost_{Large}$ in Step 9 of Algorithm 1.

TABLE II
THE PREDICTION ACCURACY OF MODE PREDICTION BASED ON THE DIFFERENCES OF SUB-BLOCKS' MVs. THREE CONDITIONS (C_1, C_2 AND C_3) USED IN THE TEST ARE GIVEN IN EQUATIONS 4, 5 AND 6.

sequence	C_1 (%)	C_2 (%)	C_3 (%)	$!(C_1 C_2 C_3)$ (%)
football(cif)	94.5	38.6	45.9	44.6
flower(cif)	92.7	71.6	59.6	58.9
foreman(cif)	97.8	77.3	71.6	17.2
mobile(cif)	87.7	67.8	60.6	41.9
akiyo(qcif)	98.9	78.8	95.2	33.7
mother-daughter(qcif)	99.1	81.3	67.5	24.7

In order to improve the prediction accuracy for small-block-size modes, we propose a metric of neighboring large block count, which is introduced in Section II-E.

E. Neighboring Large Block Count

The MV-difference-based method yields relatively low prediction accuracy for small-block-size modes (see Table II), meaning that it often predicts large-block-size modes as small-block-size modes by mistake. To address this issue, for blocks whose modes are predicted as small-block-size modes, we should check whether it is also necessary to consider large-block-size modes for them. For that purpose, a value of

Algorithm 1 Efficient Mode Decision Algorithm

- 1) Build the histograms of the current block and the co-located block in the previous frame, respectively, and compute the difference, D_{hist} , between the two histograms.
 - 2) If $D_{hist} \leq D_{low}$, then
 - a) the candidate modes are SKIP and $INTER_{16 \times 16}$,
 - b) go to step 10.
 - 3) If $D_{low} < D_{hist} \leq D_{high}$, then
 - a) the candidate modes are SKIP, $INTER_{16 \times 16}$, $INTER_{16 \times 8}$ and $INTER_{8 \times 16}$,
 - b) go to step 10.
 - 4) Subdivide the current 16×16 block into four 8×8 sub-blocks and perform motion estimation for each, obtaining the set of MVs, $\{MV_0, MV_1, MV_2, MV_3\}$; compute the block's R-D cost under this subdivision, $RDCost_{8 \times 8}$.
 - 5) If $Diff_4(MV_0, MV_1, MV_2, MV_3) < T_4$, then
 - a) the candidate modes are SKIP, $INTER_{16 \times 16}$, $INTER_{16 \times 8}$ and $INTER_{8 \times 16}$,
 - b) compute the least R-D cost of them, $RDCost_{Large}$, and go to step 9.
 - 6) If $Diff_2(MV_0, MV_1) < T_2$ or $Diff_2(MV_2, MV_3) < T_2$, then
 - a) the candidate mode is $INTER_{16 \times 8}$,
 - b) compute its R-D cost recorded as $RDCost_{Large}$, and go to step 9.
 - 7) If $Diff_2(MV_0, MV_2) < T_2$ or $Diff_2(MV_1, MV_3) < T_2$, then
 - a) the candidate mode is $INTER_{8 \times 16}$,
 - b) computed its R-D cost recorded as $RDCost_{Large}$, and go to step 9.
 - 8) Add the small-block-size modes to the candidate list; compute the value of neighboring large block count (NLBC), V_{NLBC} , and, if $V_{NLBC} > T_{NLBC}$, add the large-block-size modes into the candidate list; go to step 10.
 - 9) If $RDCost_{Large} < RDCost_{8 \times 8}$, go to step 10; otherwise, add the small-block-size modes to the candidate mode list.
 - 10) Compute the R-D cost(s) of the candidate mode(s) and pick up the best mode, *i.e.*, the mode with the least R-D cost.
-

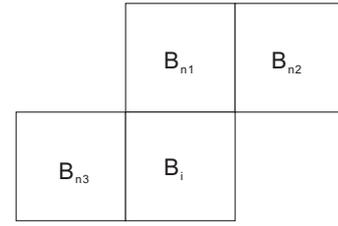


Fig. 4. Block B_i and its neighboring blocks.

neighboring large block count, V_{NLBC} , is computed for the current block and, if V_{NLBC} is above a threshold, T_{NLBC} , we should also consider large-block-size modes as candidates.

We denote the neighboring blocks of the current block, B_i , as B_{n1} , B_{n2} and B_{n3} (as illustrated in Fig. 4), use a value, $V_j, j \in \{0, 1, 2, 3\}$, to indicate the mode type for each of those four blocks, and use a value, V_t , to indicate the mode type for the co-located block with respect to B_i in the previous frame. V_j and V_t are assigned as

$$\begin{cases} 0 & \text{if the mode is a small-block-size mode} \\ 1 & \text{if the mode is } INTER_{16 \times 8} \text{ or } INTER_{8 \times 16} \\ 2 & \text{if the mode is } INTER_{16 \times 16} \text{ or SKIP} \end{cases}$$

Then, we compute V_{NLBC} as follows

$$V_{NLBC} = \sum_{j=1}^3 V_j + V_t. \quad (7)$$

Table III shows the percentage of large-block-size-mode blocks that have $V_{NLBC} > T_{NLBC}$. From this table, we see that the NLBC values of around 90% of the large-block-size-mode blocks are above the threshold, $T_{NLBC} = 4$. This fact supports our approach in correcting the false small-block-size mode predictions based on the NLBC values.

TABLE III
THE PERCENTAGE OF LARGE-BLOCK-SIZE-MODE BLOCKS WITH
 $V_{NLBC} > T_{NLBC}$.

Sequences	$T_{NLBC} = 4(\%)$
football(cif)	84.06
foreman(cif)	92.15
akiyo(qcif)	96.54
mother-daughter(qcif)	95.74
hall(qcif)	95.85
salesman(qcif)	93.87

III. EXPERIMENTAL RESULTS

A. Data Set and Computing Environment

The proposed mode decision algorithm is tested on several video sequences including football(cif), foreman(cif), akiyo(qcif), mother-daughter(qcif), hall(qcif) and salesman(qcif). Representative frames in these videos are given in Fig. 5. The algorithm is implemented based on the reference software JM17.2 [9] main profile. The simulation platform is Microsoft Windows, Intel (R) Core (TM)2 duo CPU E7500 @2.93G

TABLE IV
EXPERIMENTAL RESULTS WITH TWO ALGORITHMS (*i.e.*, (A)IVANOV'S METHOD [10] AND (B)OUR ALGORITHM) FOR SIX VIDEO SEQUENCES.

Sequence	QP	Δ PSNR(dB)		Δ B(%)		Δ T(%)	
		(a)	(b)	(a)	(b)	(a)	(b)
foreman(cif)	24	-0.00	-0.146	+0.9	+12.6	-2.4	-74.2
	28	-0.1	-0.141	+0.31	+10.8	-7.5	-69.8
	32	-0.02	-0.03	+1.8	+9.8	-13.9	-65.2
	40	-0.00	-0.06	+0.9	+8.9	-18.2	-66.0
bus(cif)	24	-0.00	-0.082	+0.4	+6.76	-5.7	-53.2
	28	-0.00	-0.069	+1.3	+8.81	-8.5	-56.6
	32	-0.03	-0.072	-0.04	+6.2	-11.3	-57.0
	40	-0.2	-0.040	-0.6	+5.7	-19.6	-56.1
flower(cif)	24	-0.00	-0.068	+0.07	+3.71	-0.10	-83.4
	28	-0.01	-0.065	+1.3	+9.07	-9.2	-81.9
	32	-0.00	-0.072	-0.59	+10.7	-17.8	-80.7
	40	-0.00	-0.081	-0.47	+9.4	-13.4	-64.6
salesman(qcif)	24	-0.02	-0.129	+2.0	+15.8	+2.2	-85.1
	28	-0.00	-0.115	+1.8	+15.7	-5.2	-84.7
	32	-0.00	-0.085	+2.3	+12.1	-15.4	-83.5
	40	+0.04	-0.084	+1.7	-2.77	-26.8	-72.0
akiyo(qcif)	24	-0.00	-0.185	+4.9	+11.9	-10.4	-84.3
	28	-0.2	-0.125	+0.02	+9.61	-14.5	-81.1
	32	-0.1	-0.117	+1.0	+5.12	-22.8	-76.6
	40	+0.00	-0.084	+0.8	+2.7	-37.2	-72.0
mother-daughter(qcif)	24	-0.091	-0.133	+5.4	+13.2	-8.7	-78.0
	28	-0.00	-0.121	+0.5	+13.1	-15.6	-78.8
	32	-0.10	-0.102	+1.1	+9.39	-24.7	-78.7
	40	-0.00	-0.066	+1.4	-0.28	-32.7	-75.5

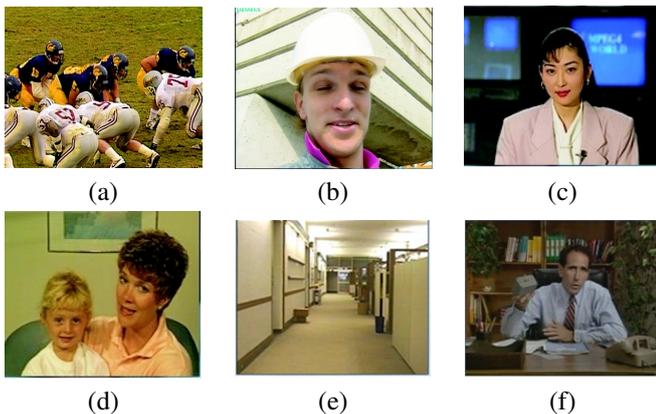
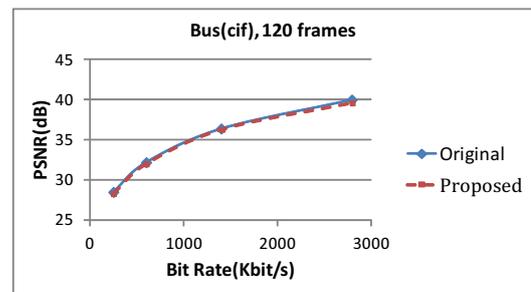


Fig. 5. Representative frames from the test video sequences: (a) football, (b) foreman, (c) akiyo, (d) mother-daughter, (e) hall, and (f) salesman.

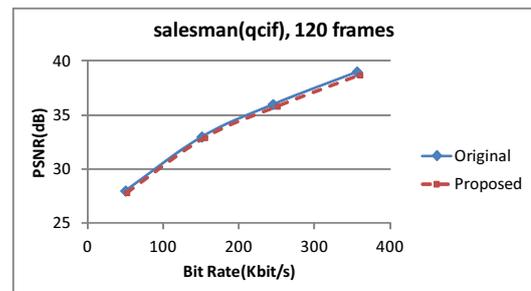
Hz with 4 GB RAM. For each sequence, 120 frames are encoded with the GOP structure,IPPP. The entropy coding method of CABAC is used, we use the previous one frame as the reference frame and we encode at the rate of 30 frames per second. In our experiments, we use the following parameter settings, $D_{low} = 20$, $D_{high} = 50$, $T_4 = 6$, $T_2 = 4$, and $T_{NLBC} = 4$.

B. Simulation Results and Analysis

Our proposed algorithm was implemented within the JM 17.2 [9]. In addition, we tested Ivanov's method [10]. The simulation results are shown in Table IV, where Δ PSNR, Δ T and Δ B mean changes in PSNR, running time and bit rate, respectively, with "+" meaning increase and "-" meaning decrease.



(a)



(b)

Fig. 6. RD curves of the original method [9] and the proposed method in (a)Bus, (b)Salesman

From Table IV, we see that, comparing with the reference software JM 17.2 [9], the proposed method is much more efficient than Ivanov's method [10] with slight decline in PSNR. In terms of coding bit rate, however, the Ivanov's method [10] slightly outperforms our algorithm. The increase of bit rate resulted from our algorithm's prediction error. Fig. 6 shows the RD curves of the original method and the proposed

method in (a)Bus and (b)Salesman under IPPP-GOP structure. The RD curves demonstrate that the RD performance of the proposed method is similar to that of the original method.

IV. CONCLUSION

In this paper, we have proposed a fast mode decision algorithm which takes advantage of the spatio-temporal coherency of the local neighborhood including and around the current block to reduce the range for the best mode search. Specifically, we propose to utilize the temporal variation of block histograms, the sub-blocks' motion coherency, the neighboring large block count metric and so forth to determine a list of candidate modes for the current block, from which the best one may be picked up based on their R-D costs. As demonstrated by the simulation results, our algorithm significantly improves the computing efficiency of both the H.264/AVC reference software JM 17.2 [9] and Ivanov's method [10] with slight decline in PSNR and rise in bit rate.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grants No. 61070103 and No. U1035004), the Program for New Century Excellent Talents in University (NCET) in China, Shandong Provincial Natural Science Foundation, China (Grant No. ZR2011FZ004) and the Scientific Research Foundation for the Excellent Middle-Aged and Youth Scientists of Shandong Province of China (Grant No. BS2011DX017).

REFERENCES

- [1] J. V. T. of ISO/IEC MPEG and I.-T. VCEG, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)*, 2003.
- [2] S. C. S. M. Y. Huang, B. Hsieh and L. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in h.264/avc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, Apr 2006.
- [3] H. Wang, S. Kwong, and C.-W. Kok, "An efficient mode decision algorithm for h.264/avc encoding optimization," *IEEE Transactions on Multimedia*, vol. 9, no. 4, pp. 882–888, Jun 2007.
- [4] X.Jing and L.-P. Chau, "Fast approach for h.264 inter mode decision," *Electronics Letters*, vol. 40, no. 17, pp. 1050–1052, Aug 2004.
- [5] D.Wu, F.Pan, K.P.Lim, S.Wu, Z.G.Li, X.Lin, S.Rahardja, and C.C.Ko, "Fast intermode decision in h.264/avc video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 953–958, Jul 2005.
- [6] B.-G. Kim, "Novel inter-mode decision algorithm based on macroblock (mb) tracking for the p-slice in h.264/avc video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 273–279, Feb 2008.
- [7] S.-H. Pi, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an h.264/avc encoder using mode and lagrangian cost correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 302–306, Feb 2009.
- [8] H. Zeng, C. Cai, and K.-K. Ma, "Fast mode decision for h.264/avc based on macroblock motion activity," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 4, pp. 491–499, Apr 2009.
- [9] "Joint video team software-jm17.2," <http://iphome.hhi.de/suehring/tml/download/>.
- [10] Y. V. Ivanov and C. J. Bleakley, "Real-time h.264 video encoding in software with fast mode decision and dynamic complexity control," *ACM Trans. Multimedia Comput. Commun. Applicat.*, vol. 6, no. 1, pp. 5:1–5:21, Feb 2010.